Journal of the Brazilian Society of Mechanical Sciences and Engineering (2023) 45:136
https://doi.org/10.1007/s40430-023-04060-w

TECHNICAL PAPER

Matrix cracking and delamination detection in GFRP laminates using pre-trained CNN models

Pankaj Chaupal[1] · S. Rohit[1] · Prakash Rajendran[1]

Received: 26 September 2022 / Accepted: 24 January 2023 / Published online: 7 February 2023

Abstract

The demand for fiber-reinforced polymer composites is increasing steadily because of their superior mechanical properties, high specific strength and stiffness, and high corrosion resistance. Matrix cracking and delamination are one of the most significant kind of damage in laminated composite structures. In this research work, our main objective is to differentiate whether the randomly oriented chopped glass fiber composite laminate is undamaged or damaged (matrix cracking and delamination) using five distinct convolutional neural network (CNN) models with transfer learning techniques. A microscopic examination is conducted on a composite laminate in the thickness direction just before and after the three-point bending test at 100 µm. Thereafter, data augmentation techniques such as mirroring, rotation, affine transformation, and noise addition are performed, and a total of 13,464 images are obtained from 2 base images. Additionally, these images are given as input to five different deep CNN models, including VGG-16, ResNet-101, NasNetMobile, MobileNet-V2, and DenseNet-201. Using augmented image datasets, the pre-trained CNN models with transfer learning are trained, validated, and tested in the proportion of 70:15:15. Thereafter, comparative studies are carried out to analyze the total trainable and non-trainable parameters, computation time, training, validation, testing accuracy, and F1 score of different CNN models. VGG-16 has 138 million trainable parameters and thus requires maximum computation time. However, MobileNet-V2 has 3 million trainable parameters that needs minimum computation time. DenseNet-201, VGG-16, MobileNet-V2, and NasNetMobile converge very fast and produce minimum validation loss and maximum accuracy whereas ResNet-101 seems not to be converged easily which leads to maximum validation loss and minimum accuracy. The highest training, validation, testing, and F1 score were observed for VGG-16, NasNetMobile, MobileNet-V2, and DenseNet-201 and the lowest for the ResNet-101 CNN model. Finally, it can be concluded that the MobileNet-V2 model performed better than the other four CNN models in terms of accuracy with respect to total parameters and computation time over the other models.

Keywords Undamaged · Damaged · Laminated composite structure · Data augmentation · CNN models · Transfer learning

1 Introduction

Fiberglass composite laminates are employed in structures that require significant strength, stiffness, low weight, good fatigue resistance, impressive corrosion resistance, and design flexibility. It possesses excellent functional properties compared to traditional materials and alloys. Applications of fiberglass composites are seen in ship building, military, automotive, sports, infrastructure, furniture, oil, and energy [1–5]. Fiberglass composites accounts for more than 95% of marine applications due to its cost advantage over carbon fiber composites [6]. Because of their orthotropic properties, these materials are prone to a variety of complicated mechanisms of failure during manufacture and under in-service stress circumstances [7, 8]. Matrix cracking, debonding, delamination and fiber breakage are the most common kind of damages in laminated composite structures [9–11]. Zhou et al. [12] studied effect of three different interface angle, i.e., 0°/0°, 45°/45°, and 90°/90° on mode I delamination and found that 45°/45° interface has the greatest and 0°/0° has the lowest fracture toughness. The effects of stacking sequence on R-curve behavior and traction-separation relations in unidirectional (UD) and multi-directional (MD)

Technical Editor: João Marciano Laredo dos Reis.

✉ Prakash Rajendran
rprakash@nitt.edu

[1] Department of Mechanical Engineering, National Institute of Technology, Tiruchirappalli, Tamil Nadu 620015, India

DCB laminates were investigated by Gong et al. [13]. They observed that, as compared to MD laminates, UD laminates had the lowest fracture toughness. According to Gong et al. [14] delamination growth and mode I fracture toughness of fiber-reinforced composites can be enhanced by using z-pinning.

The most serious damage in the composite structure is the inner-ply separation or delamination. It jeopardizes the composite structural integrity and increases the likelihood of catastrophic failure [15]. To overcome social and economic loss and its disastrous consequences several local (non-destructive testing) [16–18] and global (vibration) [19–21]-based damage detection techniques are implemented. Grondel et al. [22] used ultrasonic measurements to detect barely visible impact damage and disbond in a composite wingbox. Rajendran and Srinivasan [21] applied modeshape-based method to locate damage in composite plate structure. The results of the above damage detection methods depend on the experience and the sound knowledge of the operators and it is time-consuming as well. Thus, it is desirable and required to develop automatic damage detection methods that can detect damage based on past datasets. For damage identification, a variety of machine learning-based techniques are available, including neural networks, support vector machines, regression analysis, clustering analysis, etc. However, out of the above machine learning methods, deep CNN is widely employed for automatic damage detection based on image input data [23]. Spectrogram images obtained using acoustic emission waveforms were employed by Barile et al. [24] to train a deep neural network model for online monitoring of different composite damages (matrix cracking, debonding, delamination, and fiber breakage) in a carbon fiber-reinforced polymer composite laminate. Khan et al. [25] implemented a deep CNN to predict the delamination in the composite laminate based on vibrational spectrograms. Deep CNN was used by Sikdar et al. [26] to predict the locations of different damages based on scalograms data obtained using acoustic waveforms.

In CNN, the classification of undamaged and damaged structures can be ensured by analyzing the features map obtained through simulation [27, 28]. Different types of collision damages were classified by Patil et al. [29] using deep CNN. Cha & Choi [30] detected the crack damage in a civil infrastructure using CNN based on digital image. CNN requires millions of image data to train, validate and test the deep network model. Thus data augmentation and transfer learning are implemented by most researchers to overcome the shortage of image data problems. Sikdar et al. [26] utilized image augmentation in their research work for generating training datasets. The computation cost of CNN alone is costlier. Thus, CNN with transfer learning is developed to overcome it. Tammina [31] used VGG-16 with transfer learning based on a deep CNN to classify the different types of images. Transfer learning is a method for applying previously taught model information to a new task. Classification, regression, and clustering problems can all be benefited from transfer learning. Warmuzek et al. [32] applied pre-trained transfer learning such as DenseNet-201 and VGG-16 to recognize metal alloy microstructure constituents based on morphological characteristics. They found that the accuracy of DenseNet-201 greater than VGG-16 by 5 percent. Sun et al. [33] predicted the mechanical properties i.e., stress field in fiber-reinforced polymer composites using CNN models based on 2D slices of segmented tomography images.

From the literature review, many researchers have applied the deep CNN and deep CNN with transfer learning to identify the different types of composite damages such as delaminations, damage presence and locations, undamaged and damaged structures, microstructural behavior, and stress fields based on spectrogram images, vibration spectrograms, scalograms, features map, morphological data, and tomography images, respectively. A limited work has been done using microscopic image data to classify the undamaged and damaged composite laminate using deep CNN with transfer learning. The novelty of this study is to use of the microscopic image of randomly oriented chopped GFRP composite laminate to classify the undamaged and damaged (matrix cracking and delamination) laminates using five different CNN models with transfer learning. In the current work, a microscopic image of a glass fiber composite laminate before and after the three-point bending test is fed into CNN at a magnification factor of 100 µm. To produce a large number of image datasets, several data augmentation techniques such as mirroring, rotation, affine transformation, and noise addition were employed. Five distinct pre-trained CNNs with transfer learning are used to categorize undamaged and damaged images: VGG-16, MobileNet-V2, ResNet-101, DenseNet-201, and NasNetMobile.

Above CNN models are trained, validated, and tested with 70, 15, and 15% of total image datasets, respectively. The accuracy and effectiveness of all the models are analyzed carefully. The remainder of the study is organized as follows: Sect. 2 discusses the theoretical background of CNN, i.e., architecture of CNN and nonlinear activation function. Section 3 explains the Materials and method, various data augmentation strategies, and implementation of CNN with transfer learning. Section 4 briefly describes the results and discussions, and Sect. 5 finishes by describing conclusions.

2 Theoretical background of convolution neural network (CNN) model

CNNs are the most widely used deep neural networks for solving image classification, image segmentation, and natural sentence classification problems [34–36]. At first, CNN

was used for handwritten digit classification by LeCun et al. [37]. In the beginning, it did not work well for large-scale image and video datasets due to the limiting computing power of the central processing unit (CPU)-based systems. With the development of a highly parallel programmable device known as a graphics processing unit (GPU), the classification of big-scale visual recognition has become prominent. CNNs are most efficient and preferred over artificial neural networks (ANNs) due to their higher accuracy and efficiency. ANN has limitations with data classifications and cannot be used for image classifications. CNN analyses picture inputs with multiple perceptrons and apply learnable weights and biases to different sections of the image that can be separated. One advantage of employing CNN is that it makes use of the input pictures' local spatial coherence, allowing them to have less weight because certain characteristics are shared. In terms of memory and complexity, this procedure is definitely efficient. Unlike ANN, CNN is structured in three dimensions (length, width and depth).

2.1 Architecture of CNN model

CNN's basic structure comprises an input layer, an output layer, and multiple hidden layers. Hidden layers include the convolution layer, rectified linear unit activation function (ReLU), pooling layer, dense layer (fully connected layers), and softmax layer. Figure1 shows the schematic architecture of CNN for classifying undamaged and damaged GFRP specimens. In the convolution layer, a kernel matrix is passed over the input matrix to create a feature map for the subsequent layer. We perform a mathematical operation known as convolution by sliding the Kernel matrix over the input matrix. The activation function is just applied after the convolution layer. The ReLU is chosen as an activation function in CNN over the sigmoid and tanh functions owing to its non-saturation of gradients, which improves stochastic gradient convergence [38]. If the input is a positive value, ReLU acts as a piece-wise linear activation function and returns zero.

To train CNN models, the stochastic gradient descent (SGD) approach is commonly utilized [24]. The convolution layer's feature map output has the disadvantage of retaining the particular location of features in the input. This implies that even tiny adjustments to the input image, such as cropping and rotation, will result in a whole new feature map. To address this issue, we recommend convolution layer downsampling. Downsampling is enabled by following the nonlinearity layer with a pooling layer. Pooling aids in making the representation more resistant to minor input translations. The output of the final pooling layer is provided as input to the fully connected layer at the end of a CNN. One or more of these levels may exist. If each node in the first layer is fully linked to each node in the second layer, the fully connected layers are implied. Fully connected layers are also referred to as dense layers. The softmax activation function is then applied to the CNN model output layers, yielding the multi-class distribution. Undamaged and damaged GFRP specimens may be categorized using the multi-class probability distributions.

2.2 Nonlinear activation function

An activation function is a very crucial part of a neural network and is present in the hidden layers. It is applied on input and transferred to the next layers through output. The primary work of the activation function is to introduce nonlinearity into the network. Without a nonlinear activation function, a neural network will behave exactly like a single-layer perceptron, regardless of how many layers contain in it. The nonlinear activation function includes Sigmoid, Tanh, ReLU, and softmax which is graphically represented in Fig. 2. Sigmoid function lies between 0

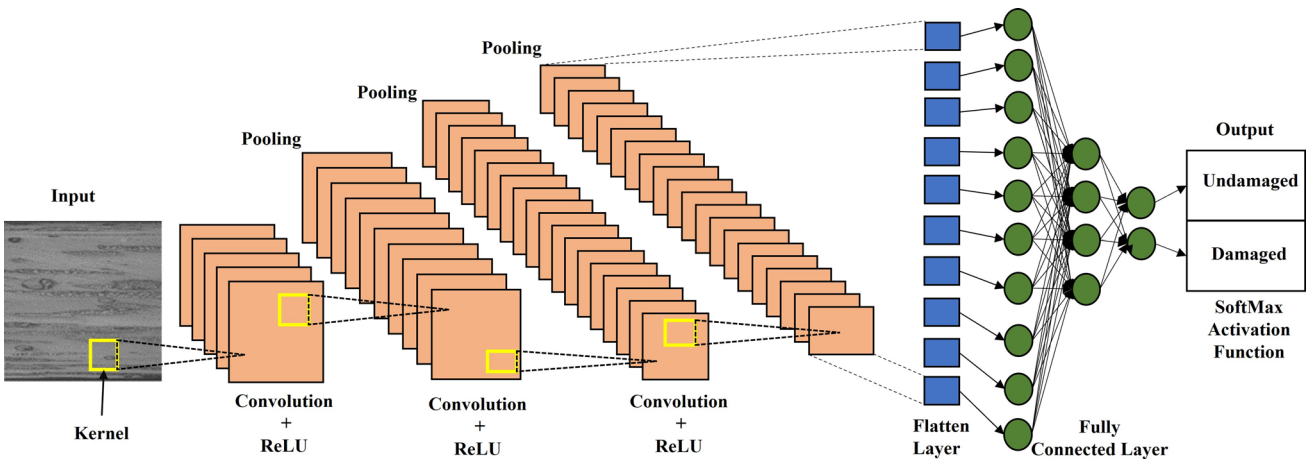

Fig. 1 CNN architecture used for classifying undamaged and damaged GFRP specimens

and 1 and is represented by Eq. (1). It is generally used in the last layer of the neural network models. It turns the model's output into a probability score, which could be easier to work on and analyze.

$$f(x) = \frac{1}{1 + \exp(-x)} \tag{1}$$

Tanh is the hyperbolic tangent function with a range of −1 to 1. It is significantly easier to represent input with highly negative, positive, or neutral values since it is zero-centered. If the output is not between 0 and 1 as given in Eq. (2), Tanh function is utilized instead of the sigmoid function. Tanh function is commonly used in recurrent neural network (RNN) for natural language processing and speech recognition tasks.

$$f(x) = \frac{2}{1 + \exp(-2x)} - 1 \tag{2}$$

The ReLU is the most popular activation function used for deep learning applications. It is faster than Sigmoid and Tanh in terms of computation time because it does not calculate exponential and division. The ReLU activation function is represented by Eq. (3).

$$f(x) = \begin{cases} 0 & \text{for } x < 1 \\ x & \text{for } x \geq 1 \end{cases} \tag{3}$$

The softmax function is an extension of the sigmoid function and is shown in Eq. (1). The major difference between both of them is a sigmoid that is used for binary classification whereas softmax is used for the multi-class classification problem. The sum of all the output in softmax is equal to unity.

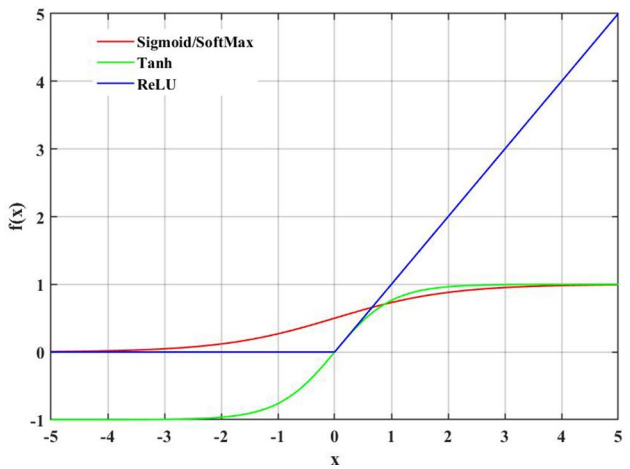

Fig. 2 Representation of nonlinear activation functions

3 Methodology to detect matrix cracking and delamination

3.1 Materials and methods

A hand layup method is employed to fabricate the GFRP composite laminate. For fabricating the composite laminate, randomly oriented chopped glass fiber of 459 GSM is chosen as the reinforcing material and isophthalic polyester resin of RPL-100 is used as matrix material. The ratio between the reinforced material and matrix material is chosen as 3:7 by weight. Additionally, the matrix material is combined in a 10:1 ratio with the hardener (methyl ethyl ketone peroxide). Further, the flat mold surface is covered with a thin layer of wax to eliminate minor surface imperfections like scratches, stains, and oxidation. Thereafter, polyvinyl alcohol (PVA) is applied as a chemical release agent over the surface to easy removal of the laminate from the mold after fabrication. The hardener and isophthalic polyester resin are mixed properly in the beaker. The mixture of resin and hardener is brushed onto the dies. A layer of randomly oriented chopped glass fabric is placed over the die and again mixture of resin and a hardener is applied over the fabric layer. The same process is repeated up to three layers. The extra resin from the laminate is removed using the cylindrical roller by squeezing. The laminates are cured at atmospheric conditions for at least 24 h after finishing the hand layup. After curing, the laminate is removed from the mold and cut into the specified dimensions for the three-point bending test as per ASTM D790 standard as shown in Fig. 3a.

The three-point bending test is conducted based on the ASTM D790 standard. Blue Star's 20kN Universal Testing Machine (UTM) is utilized to conduct the three-point bending test. Figure 3b depicts the UTM test setup and the test specimen before and after the three-point bending test. A constant feed rate of 2 mm/min is maintained during the experimental bending analysis. Before and after conducting the three-point bending test a microscopic examination is carried out on the laminate at 100 µm in the thickness direction. The microscopic image of the undamaged, damaged (matrix cracking and delamination), and test specimens is shown in Fig. 4a–c, respectively.

3.2 Data augmentation

The method for detecting undamaged and damaged GFRP specimens using CNNs is illustrated in Fig. 5. The detection of undamaged and damaged (matrix cracking and delamination) specimens using CNN requires a large number of image datasets. Thus, the microscopic image of

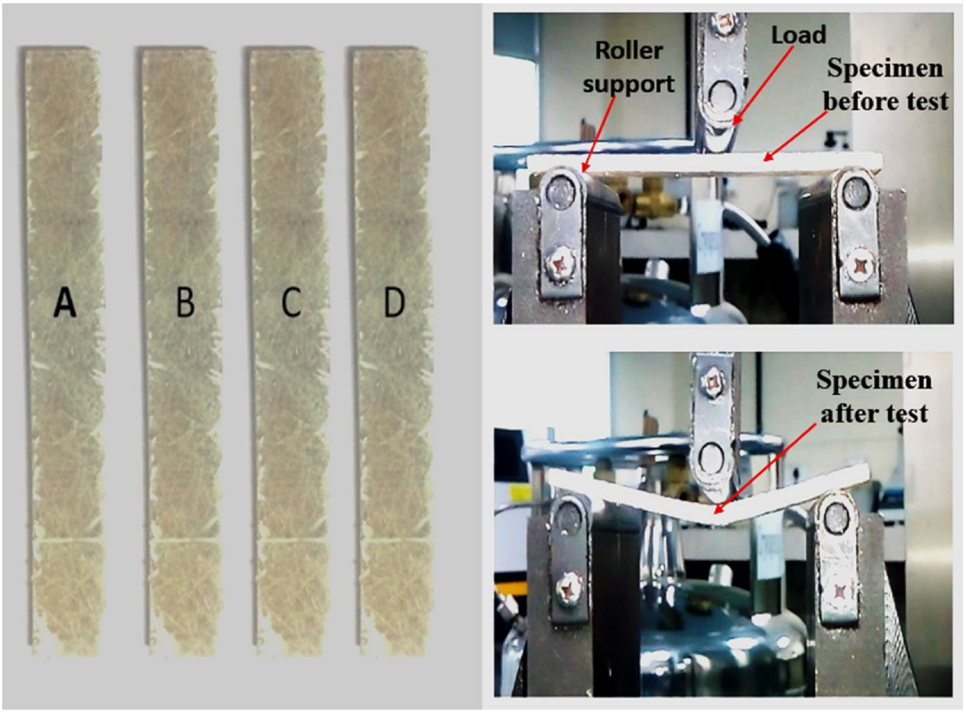

Fig. 3 **a** GFRP test specimen, **b** three-point bending test experimental setup

randomly oriented chopped glass fiber composite laminate is used for the CNN implementation as shown in Fig. 4a, b. From the literature review, it is clear that a large number of image data are required for performing CNNs. Therefore, these datasets can be artificially produced by applying data augmentation process [26, 39, 40]. Data augmentation can be done by applying mirroring, rotation, affine transformation, and adding artificial noise to the base image dataset.

In the mirroring operation, at first, the base image data are flipped over the horizontal and vertical axes and then the vertical image is re-flipped over the horizontal axis. Each coordinate of the base dataset is reconstructed corresponding to the mirror line with equal distance in the opposite direction. Mirroring is also referred to as reflection. Thus, with the help of a mirroring operation, four image data are obtained for each undamaged and damaged case. Further, each mirrored image dataset is rotated between 1°-360° with a step size of 1° degree per iteration. Basically, in the rotation, each image data is rotated about a fixed point either in the clockwise or anti-clockwise direction. Hence, the total number of images generated after the rotation operation for each undamaged and damaged case were found to be 1440. Thereafter, an affine transformation is performed over the four mirrored images with a step size of 0.02 and a total of 800 images are obtained. Affine transformation is a type of linear programming method that preserves parallel sets of points, lines, and planes that remain parallel after transformation. After completion of mirroring, rotation and affine transformation process speckle noise and Gaussian noise with mean and variance of 0.004 are applied separately to the total number of images obtained after transformation, i.e., 2244. Finally, the total number of images obtained after the successful completion of data augmentation and noise addition for each undamaged and damaged case is 6732. The undamaged and damaged images after transformation are shown in Fig. 6.

The data augmentation and noise addition simulation are performed in MATLAB®2022a. Once an actual dataset is generated, the five different CNN models with transfer learning are applied to train, validate, and test the models with 70, 15, and 15%, respectively. For solving the deep CNN problem with the huge number of image inputs a highly configured GPU system is required. Thus, the deep CNN models such as VGG-16, MobileNet-V2, ResNet-101, DenseNet-201, and NasNetMobil are simulated in "Google Colab" with 25.46 GB RAM, a python 3 and Google Computed Engine backend (GPU).

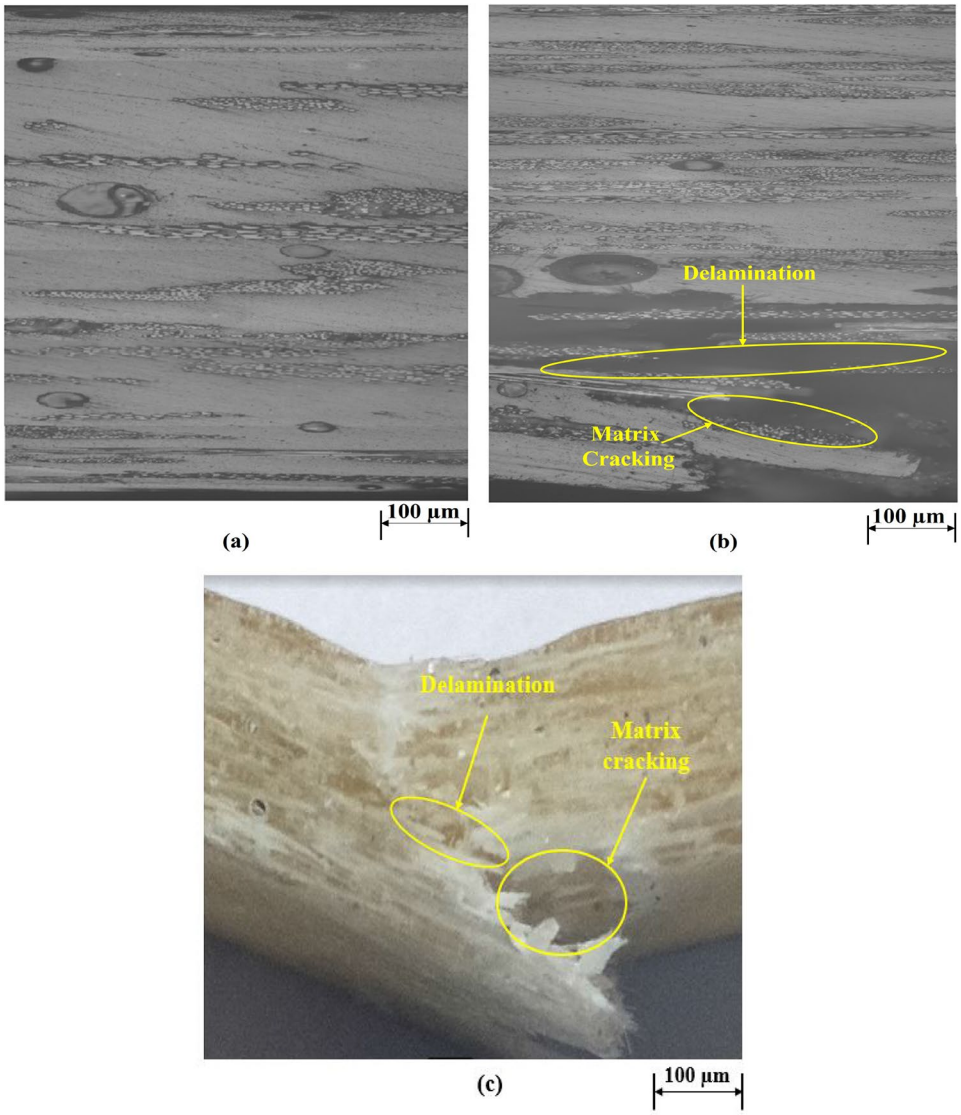

Fig. 4 Microscopic image of GFRP composite laminate at 100 μm. **a** Undamaged specimen, **b** damaged specimen, **c** test specimen

3.3 Implementation of CNN with transfer learning

In this research implementation transfer learning with five different CNN models such as VGG-16, MobileNet-V2, ResNet-101, DenseNet-201, and NasNetMobile are used for classifying undamaged and damaged GFRP composite specimens. Transfer learning is the process of reusing previously learned CNN to tackle a new classification issue. It lowers the cost of model training while increasing the efficacy of the machine learning model. It is used in machine learning models when there is a lack of labeled data. By changing the size of the convolution layer filter and class out in the pre-trained CNN model a new classification problem can be implemented. VGG-16 architecture was developed by Karen and Andrew in 2014 [31, 41]. VGG 16 has 21 layers in total, with 13 convolutional layers, 5 max pooling layers, and 3 dense fully connected layers. Out of the 21 mentioned layers, only 16 (13 convolutional layers and 3 fully connected layers) have learnable parameters, i.e., which are subjected to change during the models training to optimize the accuracy. However, max pooling layers have only one filter with a fixed size of 2x2, and hence, its hyperparameters do not change as the model's training is performed. The detailed architecture of VGG-16 obtained after the rigorous simulation in python using Google Colab is summarized in Table 1. After applying transfer learning to the above VGG-16, undamaged and damaged classification can be obtained.

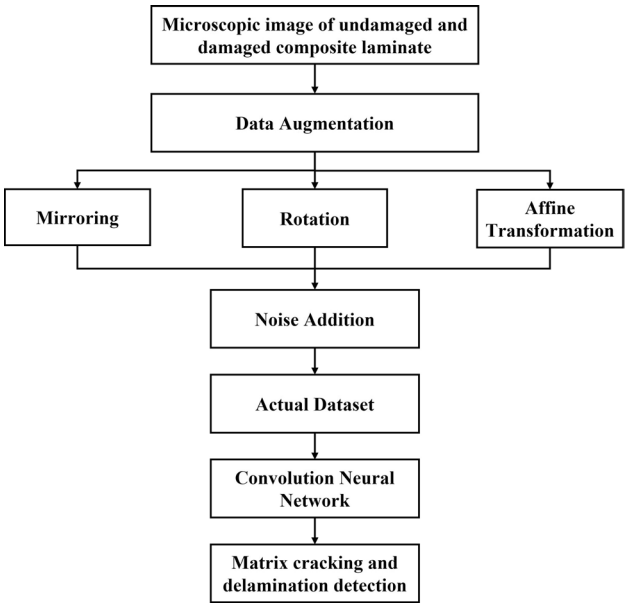

Fig. 5 Methodology for detecting undamaged and damaged composite laminate

The obtained architecture of MobileNet-V2, ResNet-101, DenseNet-201, and NasNetMobile using python and Google Colab are highly complex which cannot be represented easily. The earlier research work which explains the architecture of the remaining CNNs model is explained below. MobileNetV2 developed by Google in 2017, contains convolution layers followed by 19 bottleneck layers and a classification layer. These 19 bottleneck layers are used for extracting features based on the classification needed. It employs a linear bottleneck with an inverted residual block which increases performance over MobileNetV1. The architecture of MobileNetV2 can be seen in the following literature [42]. The ResNet design was influenced using VGG-19 model which was developed by Kaiming et al. [43]. ResNet-101 is one of ImageNet's most in-depth designs (Image classification). It comprises 101 layers which is a modified version of the 50-layer ResNet. It is based on the notion of residual learning that is viewed as the subtraction of input characteristics learned from related layers. The architecture of ResNet-101 was explained by Ghosal et al. [44] for brain tumor classification. Gao Huang created the DenseNet-201 convolutional neural network. It comprises of 201 layers which has a more dense connection than VGG-16 and ResNet-101. DenseNet overcomes the vanishing-gradient problem that increases feature map propagation and reduces the number of parameters [45]. The architecture of DenseNet-201 can be understood by following reference [46]. The Google Brain team developed the Neural Architecture Search Network (NASNet); it has two primary

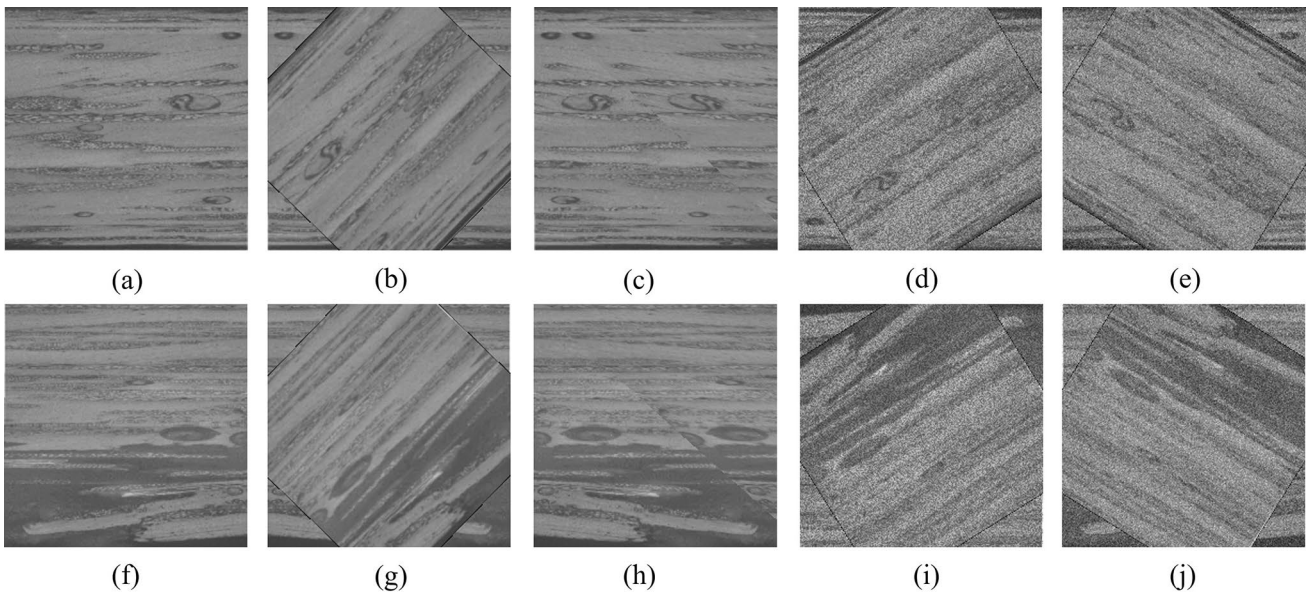

Fig. 6 Transformed undamaged specimen at 100μm (**a** mirroring, **b** rotation, **c** affine, **d** speckle noise, **e** Gaussian noise), damaged (**f** mirroring, **g** rotation, **h** affine, **i** speckle noise, **j** Gaussian noise images)

Table 1 Configuration of VGG-16 CNN model

Name of the layer	Layer description (output shape)
Input layer	$224 \times 224 \times 3$, microscopic image at 100 μm
Conv1, Conv2	Convolution filter = 224×224, number of filters = 64,
	Convolution filter = 224×224, number of filters = 64
Pool	Maximum pooling = 112×112, number of filters = 64
Conv1, Conv2	Convolution filter = 112×112, number of filters = 128,
	Convolution filter = 112×112, number of filters = 128
Pool	Maximum pooling = 56×56, number of filters = 128
Conv1, Conv2, Conv3	Convolution filter = 56×56, number of filters = 256,
	Convolution filter = 56×56, number of filters = 256,
	Convolution filter = 56×56, number of filters = 256
Pool	Maximum pooling = 28×28, number of filters = 256
Conv1, Conv2, Conv3	Convolution filter = 28×28, number of filters = 512,
	Convolution filter = 28×28, number of filters = 512,
	Convolution filter = 28×28, number of filters = 512
Pool	Maximum pooling = 14×14, number of filters = 512
Conv1, Conv2, Conv3	Convolution filter = 14×14, number of filters = 512,
	Convolution filter = 14×14, number of filters = 512,
	Convolution filter = 14×14, number of filters = 512
Pool	Maximum pooling = 7×7, number of filters = 512
Flatten	Output = 25088
FC layer-1 (dense layer-1)	Output = 4096
FC layer-2 (dense layer-1)	Output = 4096
Softmax	Output = 1000

functions: normal cell and reduction cell [47]. It is based on the principle of reinforcement learning. The ResNet design is made up of two components: the controller recurrent neural network (CRNN) and CNN. The NASNet architecture is divided into two variants based on the input picture size: NASNetLarge (331×331) and NASNetMobile (224×224). Based on the total number of parameters, NASNetMobile outperforms NASNetLarge [48]. The NASNetMobile architecture can be understood by following research work [48]. The input used for all the above CNN models was an RGB image with $224 \times 224 \times 3$ dimensions. Where 224, 224, and 3 represent the width, height, and channel, respectively. Each of the three channels that make up a pixel indicates a different color. Red, green, and blue are the three channels that make up a color image. To improve the aforementioned CNN model, a stochastic gradient descent optimizer with a learning rate of 20 is employed. With a batch size of 70 for 30 epochs, the full image data set is divided in the proportion of 70:15:15 for training, validation, and testing, respectively.

Table 2 Comparison of pre-trained deep CNN based on parameters

CNN model	Total parameters	Trainable parameters	Non-trainable parameters	Computation time (in s)
VGG-16	138,357,544	138,357,544	0	4310.7
ResNet-101	44,707,176	44,601,832	105,344	3965.8
NasNetMobile	5,326,716	5,289,978	36,738	2475.9
MobileNet-V2	3,538,984	3,504,872	34,112	791.4
DenseNet-201	26,242,984	20,013,928	229,056	2661.6

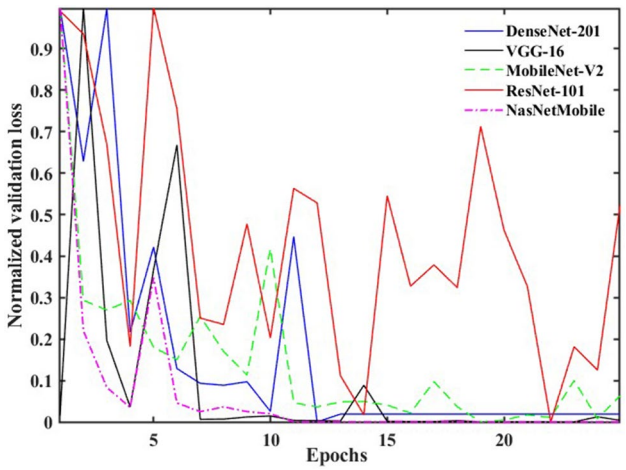

Fig. 7 A typical validation loss of the proposed different CNN models

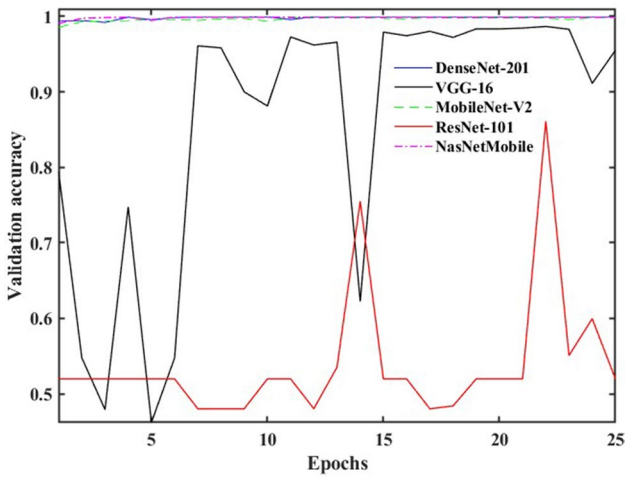

Fig. 8 A typical validation accuracy of the proposed different CNN models

The number of total parameters, trainable parameters and non-trainable parameters for all the CNN models is explained in Table 2. The memory size is required to simulate the CNN models depends on the total number of parameters. Disk space of 515 MB is required to simulate 138 million parameters [49], i.e., to train a VGG-16 CNN model. VGG-16 has 138 million parameters which requires 4311 seconds to train the network model. Similarly, ResNet-101, NasNetMobile, MobileNet-V2, and DenseNet-201 have 44, 5, 3, and 26 million parameters which require 3966, 2480, 791, and 2662 s to train the model, respectively. VGG-16 requires maximum computation time and MobileNet-V2 needs minimum computation time.

4 Results and discussion

4.1 Validation loss and validation accuracy

Figures 7 and 8 depict the validation loss and validation accuracy versus the number of epochs of the DenseNet-201, VGG-16, MobileNet-V2, ResNet-101, and NASNetMobile. The network is trained, validated and tested with 9424, 2020, and 2020 images, respectively, for 25 epochs with a batch size of 32 per epoch without overfitting the datasets. The steps per epoch of 295 are used to train the batches of samples. Further, the validation loss is normalized and plotted because they observed a huge difference between the successive loss among the different models.

The validation loss for the CNN models of DenseNet-201, VGG-16, MobileNet-V2, and NasNetMobile starts converging at 12, 7, 11, and 11 epochs, respectively, and has a minimum validation loss, whereas the other CNN models, ResNet-101, seem not to be converged that have maximum validation loss as shown in Fig. 7. Similarly, the validation accuracy for the CNN models such as DenseNet-201, MobileNet-V2, and NasNetMobile converges very fast i.e., within 5 epochs and VGG-16 converges approximately at 15 epochs and having the maximum accuracy as shown in Fig. 8. However, ResNet-101 tends to be diverged and yield lower accuracy. Therefore, the optimum number of epochs for DenseNet-201, VGG-16, MobileNet-V2, NasNetMobile, and ResNet-101 are chosen as 12, 7, 11, 11, and 25 epochs, respectively. Thus, the selection of an optimum number of epoch sizes can minimize the computation time and cost of validation and testing.

4.2 Accuracy comparison of different CNN models

Table 3 represents the training, validation, testing accuracy and F1 score of different CNN models. The training,

Table 3 Training, validation, and testing accuracy of different CNN models

CNN model	Training accuracy (in %)	Validation accuracy (in %)	Testing accuracy (in %)	F1 score
VGG-16	99.71	99.50	99.60	99.60
ResNet-101	66.30	84.16	84.11	83.06
NasNetMobile	100.00	100.00	99.95	99.95
MobileNet-V2	100.00	99.90	100.00	100.00
DenseNet-201	100.00	99.95	100	100.00

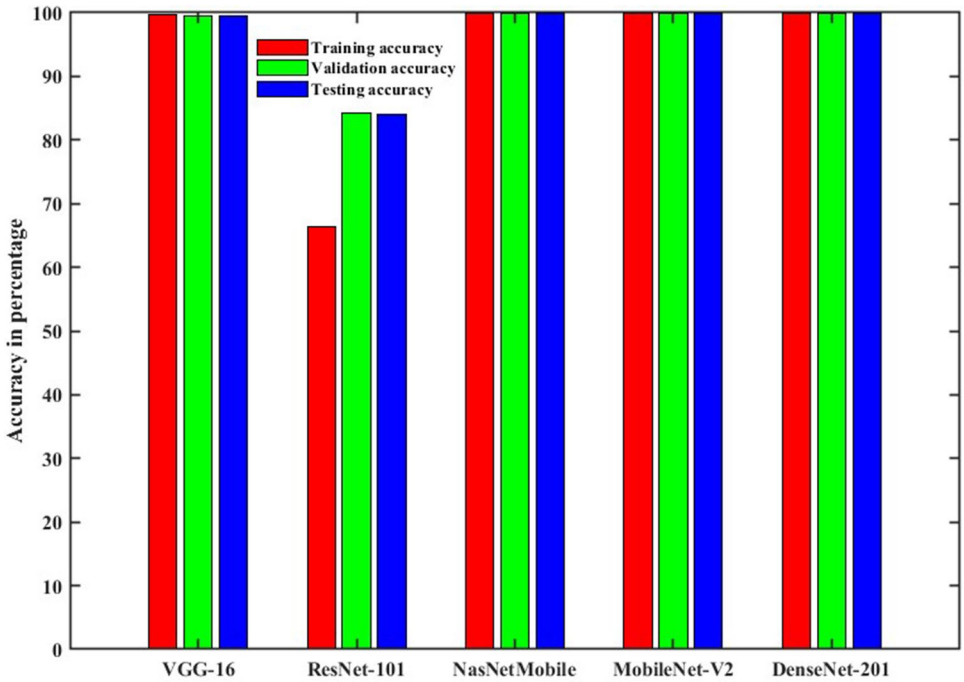

Fig. 9 Comparison among training, validation and testing the accuracy of the five different CNN models

validation, and testing accuracy of VGG-16, NasNetMobile, MobileNet-V2, and DenseNet-201 have obtained greater than 99 percent and ResNet-101 has accuracy below 85 percent. Similarly, the F1 score is higher for VGG-16, NasNetMobile, MobileNet-V2, and DenseNet-201, i.e., greater than 99 percent and lower for ResNet-101 i.e., less than around 85 percent, respectively. Here, F1 basically represents the harmonic mean of precision and recall. The highest F1 score shows the greatest precision and recall. Thus, the CNN models such as VGG-16, NasNetMobile, MobileNet-V2, and DenseNet-201 are performed better as compared to ResNet-101 CNN model as shown in Fig. 9.

4.3 Prediction accuracy of different CNN models

For better understanding of the CNN models performance, the confusion matrix was calculated on each test set. Figure 10a–e shows the confusion matrix for the CNN models such as VGG-16, ResNet-101, NasNetMobile, MobileNet-V2, and DenseNet-201, respectively. The total number of images used for testing the model was 15 percent of the entire dataset, i.e., 2020. The trained model has to classify the dataset into undamaged and damaged cases. Figure 10 illustrates that VGG-16, ResNet-101, NasNetMobile, MobileNet-V2, and DenseNet-201 classify 99.60, 85.10, 99.95 100, and 100.00% correct values, respectively, whereas 0.40 15.88, 0.05, 0.00, and 0.00% incorrect values, respectively. The overall prediction accuracy of VGG-16, NasNetMobile, MobileNet-V2, and DenseNet-201 models shows greater than ResNet-101 model.

5 Conclusion

Laminated composites are prone to numerous complex damages such as matrix cracking and delamination. In this research article, five different CNN models such as VGG-16, ResNet-101, NasNetMobile, MobileNet-V2, and DenseNet-201 with transfer learning are proposed to predict the undamaged and damaged (matrix cracking and delamination) in a randomly oriented chopped GFRP composite laminate based on the macroscopic image at 100 µm before and after the three-point bending test. Four different data augmentation techniques such as mirroring, rotation, affine transformation, and noise addition are performed and a total of 13,464 images are obtained from the two base images. The input image used for all the CNN models was RGB with $224 \times 224 \times 3$ dimensions. The entire dataset is divided into training, validation, and testing in the proportion of 70:15:15, respectively. A stochastic gradient descent optimizer with a learning rate of 20 is selected to optimize the models. The entire datasets are trained, validated, and tested with a batch size of 32, steps per epoch of 295 for 25 epochs. The training, validation, testing accuracy, and F1 score of VGG-16, NasNetMobile, MobileNet-V2 and DenseNet-201 models which are performed better than ResNet-101 model by approximately 15 percent. VGG-16 CNN model has highest parameters, i.e., 138 million, hence requires the maximum computation time. However, MobileNet-V2 has the lowest parameters i.e., 3 million, thus it requires the lowest computation time. Finally, the overall performance of MobileNet-V2 surpass the other four CNN models.

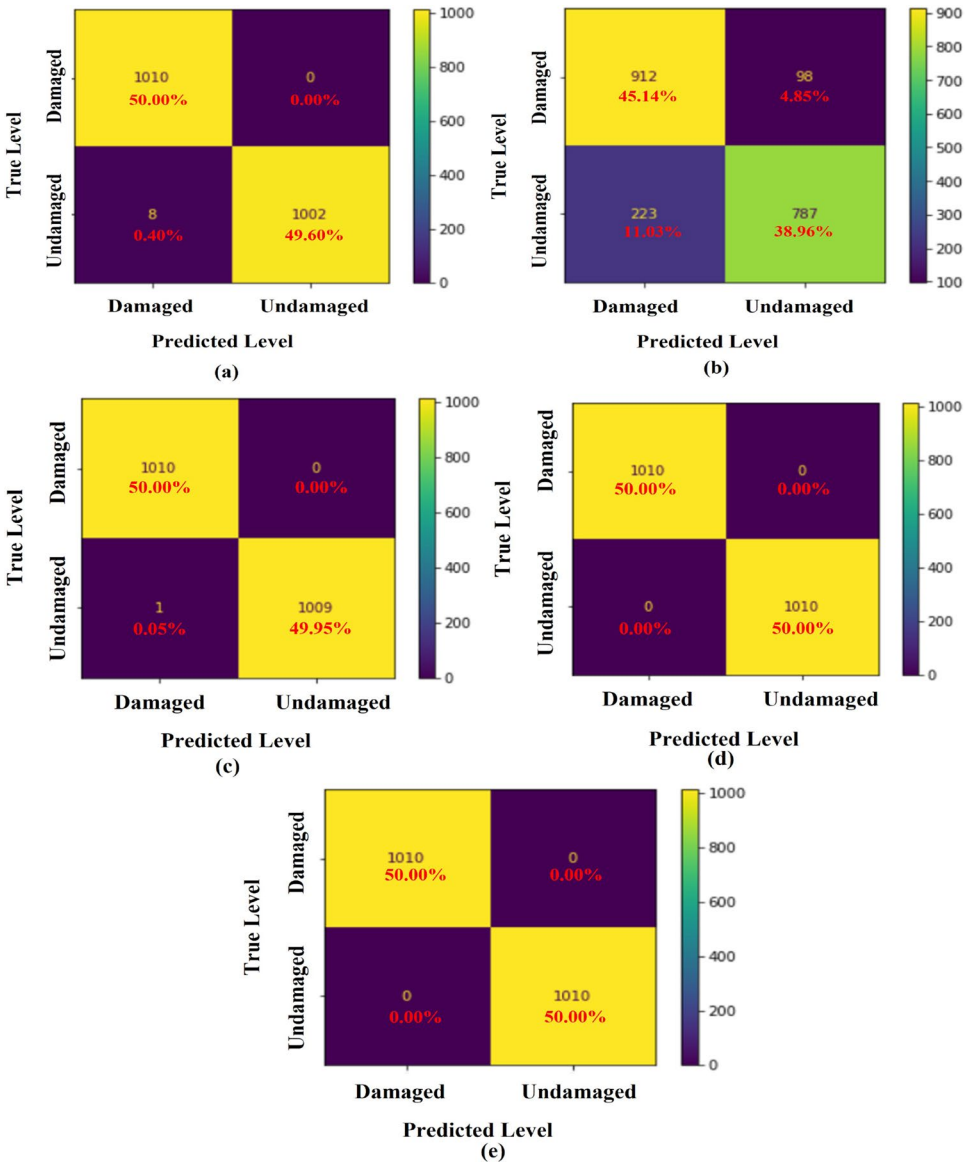

Fig. 10 Confusion matrix for different convolution neural network. **a** VGG-16, **b** ResNet-101, **c** NasNetMobile, **d** MobileNetV2, **e** DenseNet-201

Acknowledgements The authors gratefully acknowledge the support of the National Institute of Technology Tiruchirappalli, Ministry of Education, India, through SEED grant number NITT/R &C/SEED GRANT/2021-22/PROJ.NO.35.

Declarations

Conflict of interest The authors state that they have no known conflicting financial or personal interests that may have seemed to affect the work presented in this study.

References

1. Peret T, Clement A, Freour S, Jacquemin F (2017) Effect of mechanical states on water diffusion based on the free volume theory: Numerical study of polymers and laminates used in marine application. Compos B Eng 118:54–66
2. Kootsookos A, Mouritz AP (2004) Seawater durability of glass-and carbon-polymer composites. Compos Sci Technol 64(10–11):1503–1511
3. Khan A, Raouf I, Noh YR, Lee D, Sohn JW, Kim HS (2022) Autonomous assessment of delamination in laminated composites using deep learning and data augmentation. Compos Struct 290:115502
4. Chaurasia BK, Kumar D, Paswan MK (2022) Experimental studies of failure in l-shaped carbon fiber-reinforced polymer composite under pullout and four-point bending. J Inst Eng (India): Ser D. https://doi.org/10.1007/s40033-022-00411-4
5. Pastuszak PD, Muc A (2013) Application of composite materials in modern constructions. In: Key engineering materials, vol 542. Trans Tech Publ, pp 119–129

6. Shenoi R, Dodkins A (2000) Design of ships and marine structures made from FRP composite materials. Compr Compos Mater 6:429–449
7. De Luca A, Caputo F (2017) A review on analytical failure criteria for composite materials. AIMS Mater Sci 4(5):1165–1185
8. Talreja R (2013) Studies on the failure analysis of composite materials with manufacturing defects. Mech Compos Mater 49(1):35–44
9. Chaupal P, Prakash R (2022) Damage identification in composite structure using machine learning techniques based on acoustic emission waveforms. In: Recent advances in manufacturing modelling and optimization: select proceedings of RAM 2021. Springer, pp 149–158
10. Heslehurst RB (2014) Defects and damage in composite materials and structures, vol. 19. CRC Press, Boca Raton
11. Chaupal P, Kumar D (2022) Progressive damage analysis of random oriented chopped glass fiber-reinforced laminate under three-point bending test. J Inst Eng (India): Ser D. https://doi.org/10.1007/s40033-022-00410-5
12. Zou L, Gong Y, Tian D, Zhao L, Zhang J, Hu N (2023) Effect of interface angle on mode i delamination damage behavior of multidirectional fully isotropic laminates with the same global stiffness. Thin-Walled Struct 182:110211
13. Gong Y, Chen X, Li W, Zhao L, Tao J, Zhang J, Hu N (2021) Delamination in carbon fiber epoxy DCB laminates with different stacking sequences: R-curve behavior and bridging traction-separation relation. Compos Struct 262:113605
14. Gong Y, Chen X, Zou L, Li X, Zhao L, Zhang J, Hu N (2022) Experimental and numerical investigations on the mode i delamination growth behavior of laminated composites with different z-pin fiber reinforcements. Compos Struct 287:115370
15. Banks-Sills L (2018) Interface fracture and delaminations in composite materials. Springer
16. Ni Q-Q, Hong J, Xu P, Xu Z, Khvostunkov K, Xia H (2021) Damage detection of CFRP composites by electromagnetic wave nondestructive testing (EMW-NDT). Compos Sci Technol 210:108839
17. Gryzagoridis J, Findeis D (2010) Impact damage detection on composites using optical ndt techniques. Insight-Non-Destr Test Cond Monit 52(5):248–251
18. Amaro A, Reis P, De Moura M, Santos J (2012) Damage detection on laminated composite materials using several ndt techniques. Insight-Non-Destr Test Cond Monit 54(1):14–20
19. Das S, Saha P, Patro S (2016) Vibration-based damage detection techniques used for health monitoring of structures: a review. J Civ Struct Heal Monit 6(3):477–507
20. Avci O, Abdeljaber O, Kiranyaz S, Hussein M, Gabbouj M, Inman DJ (2021) A review of vibration-based damage detection in civil structures: from traditional methods to machine learning and deep learning applications. Mech Syst Signal Process 147:107077
21. Rajendran P, Srinivasan SM (2016) Identification of added mass in the composite plate structure based on wavelet packet transform. Strain 52(1):14–25
22. Grondel S, Assaad J, Delebarre C, Moulin E (2004) Health monitoring of a composite wingbox structure. Ultrasonics 42(1–9):819–824
23. Khan A, Kim N, Shin JK, Kim HS, Youn BD (2019) Damage assessment of smart composite structures via machine learning: a review. JMST Adv 1(1):107–124
24. Barile C, Casavola C, Pappalettera G, Kannan VP (2022) Damage monitoring of carbon fibre reinforced polymer composites using acoustic emission technique and deep learning. Compos Struct 292:115629
25. Khan A, Ko D-K, Lim SC, Kim HS (2019) Structural vibration-based classification and prediction of delamination in smart composite laminates using deep learning neural network. Compos B Eng 161:586–594
26. Sikdar S, Liu D, Kundu A (2022) Acoustic emission data based deep learning approach for classification and detection of damage-sources in a composite panel. Compos B Eng 228:109450
27. Gulgec NS, Takáč M, Pakzad SN (2017) Structural damage detection using convolutional neural networks. In: Model validation and uncertainty quantification, vol 3. Springer, pp 331–337
28. Cofre-Martel S, Kobrich P, Lopez Droguett E, Meruane V (2019) Deep convolutional neural network-based structural damage localization and quantification using transmissibility data. Shock Vib 2019:1–27
29. Patil K, Kulkarni M, Sriraman A, Karande S (2017) Deep learning based car damage classification. In: 2017 16th IEEE international conference on machine learning and applications (ICMLA). IEEE, pp 50–54
30. Cha Y-J, Choi W, Büyüköztürk O (2017) Deep learning-based crack damage detection using convolutional neural networks. Comput-Aided Civil Infrastruct Eng 32(5):361–378
31. Tammina S (2019) Transfer learning using vgg-16 with deep convolutional neural network for classifying images. Int J Sci Res Publ (IJSRP) 9(10):143–150
32. Warmuzek M, Żelawski M, Jałocha T (2021) Application of the convolutional neural network for recognition of the metal alloys microstructure constituents based on their morphological characteristics. Comput Mater Sci 199:110722
33. Sun Y, Hanhan I, Sangid MD, Lin G (2020) Predicting mechanical properties from microstructure images in fiber-reinforced polymers using convolutional neural networks. arXiv preprint arXiv:2010.03675
34. Krizhevsky A, Sutskever I, Hinton GE (2017) Imagenet classification with deep convolutional neural networks. Commun ACM 60(6):84–90
35. Chen Y (2015) Convolutional neural network for sentence classification. Master's thesis, University of Waterloo
36. Farabet C, Couprie C, Najman L, LeCun Y (2012) Learning hierarchical features for scene labeling. IEEE Trans Pattern Anal Mach Intell 35(8):1915–1929
37. LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. Proc IEEE 86(11):2278–2324
38. Krizhevsky A, Sutskever I, Hinton GE (2017) Imagenet classification with deep convolutional neural networks. Commun ACM 60(6):84–90
39. Ewald V, Goby X, Jansen H, Groves RM, Benedictus R (2018) Incorporating inductive bias into deep learning: a perspective from automated visual inspection in aircraft maintenance. In: Proc. 10th intl symposium on NDT in aerospace, Dresden, pp 1–9
40. Ewald V, Groves RM, Benedictus R (2019) Deepshm: a deep learning approach for structural health monitoring based on guided lamb wave technique. In: Sensors and smart structures technologies for civil, mechanical, and aerospace systems 2019, vol 10970. SPIE, pp 84–99
41. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556
42. Xiang Q, Wang X, Li R, Zhang G, Lai J, Hu Q (2019) Fruit image classification based on mobilenetv2 with transfer learning technique. In: Proceedings of the 3rd international conference on computer science and application engineering, pp 1–7
43. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778
44. Ghosal P, Nandanwar L, Kanchan S, Bhadra A, Chakraborty J, Nandi D (2019) Brain tumor classification using resnet-101 based squeeze and excitation deep neural network. In: 2019 2nd

international conference on advanced computational and communication paradigms (ICACCP). IEEE, pp 1–6
45. Zhong Z, Zheng M, Mai H, Zhao J, Liu X (2020) Cancer image classification based on densenet model. In: Journal of physics: conference series, vol 1651. IOP Publishing, p 012143
46. Jaiswal A, Gianchandani N, Singh D, Kumar V, Kaur M (2021) Classification of the Covid-19 infected patients using densenet201 based deep transfer learning. J Biomol Struct Dyn 39(15):5682–5689
47. Addagarla SK, Chakravarthi GK, Anitha P (2020) Real time multi-scale facial mask detection and classification using deep transfer learning techniques. Int J 9(4):4402–4408
48. Radhika K, Devika K, Aswathi T, Sreevidya P, Sowmya V, Soman K (2020) Performance analysis of nasnet on unconstrained ear recognition. In: Nature inspired computing for data science. Springer, pp 57–82
49. Khan A, Khalid S, Raouf I, Sohn J-W, Kim H-S (2021) Autonomous assessment of delamination using scarce raw structural vibration and transfer learning. Sensors 21(18):6239